UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/855,776	05/16/2001	Norichika Kumamoto	1602.1001	9123

21171 7590 12/13/2004

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 12/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/855,776	KUMAMOTO ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 May 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-28 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 10/18/2004.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Drawings

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: The examiner has been unable to find reference numbers 31c (Fig.9) and 30e (Fig.13) in the specification. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the

applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

5. Claim 1 is objected to because of the following informalities: The use of the word "being" in line 4 of the claim appears to be grammatically improper. Please reword that portion of the claim. Appropriate correction is required.

Maintained Rejections

6. Applicant has failed to overcome the prior art rejections set forth in the previous Office Action. Consequently, these rejections are respectfully maintained by the examiner and are copied below for applicant's convenience.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 1-6 and 8 are rejected under 35 U.S.C. 102(b) as being anticipated by Parady, U.S. Patent No. 5,933,627.

9. Referring to claim 1, Parady has taught a processor control apparatus for controlling a plurality of arithmetic units, said processor control apparatus comprising a plurality of instruction control units (Fig.5, components 150 and 154) issuing a series of instructions to said plurality of arithmetic units (Fig.5, components 156, 160, 168, and 170), wherein at least one of said instruction control units being operable to switch between a first execution process driving said plurality of arithmetic units by a single series of instructions and a second execution process driving said plurality of arithmetic units by a plurality of different series of instructions, respectively. See the abstract and note that while instructions belonging to a first execution process may be issued from a single instruction control unit, a long latency event (or some other event) within that process will result in switching to driving arithmetic units using a second set of instructions belonging to a second execution process.

10. Referring to claim 2, Parady has taught a processor control apparatus as described in claim 1. Parady has further taught that said at least one instruction control unit each perform a switching process switching between said first execution process and said second execution process according to information which is contained in advance in a series of instructions. See the abstract, column 3, lines 57-64. Note that execution processes are switched due to multiple events such as a load instruction missing the data cache. The load instruction is contained in advance in a series of instructions. Also, other supported instructions are jumps to threads. See column 4, line 63, to column 5, line 5. These instructions are also considered information contained in advance in the series of instruction.

11. Referring to claim 3, Parady has taught a processor control apparatus as described in claim 1. Parady has further taught that when an M-th one of said instruction control units issues

a second series of instructions to an N-th one of said arithmetic units which is performing said second execution process based on a first series of instructions issued by an N-th one of said instruction control units different from said M-th instruction control unit, said M-th instruction control unit is set in a wait state until said N-th arithmetic unit completes said second execution process. See column 4, lines 42-52, and note that if the second execution process is of highest priority (or at least higher priority than the process corresponding to the second series of instruction), then the M-th control unit will have to wait to issue instructions until the second execution process completes. Also, it should be realized that there are M instruction control units (Fig.3, components 102-108, and Fig.5, components 154) and N functional units (Fig.1 and Fig.5).

12. Referring to claim 4, Parady has taught a processor control apparatus as described in claim 1. Parady has further taught a first storage element for holding a plurality of series of instructions, wherein when an M-th one of said instruction control units issues a second series of instructions to an N-th one of said arithmetic units which is performing said second execution process based on a first series of instructions issued by an N-th one of said instruction control units different from said M-th instruction control unit, said second series of instructions from said M-th instruction control unit are stored in said first storage element, and wherein said N-th arithmetic unit executes instructions which are stored in said first storage element based on information contained in said first series of instructions issued by said N-th instruction control unit. See Fig.3, for instance, and note the instruction control unit (say 104 in combination with 28), comprises a first storage element to hold thread instructions. If the thread 0 control unit (102 in combination with 28) issues a load that misses the cache or a jump to thread instruction,

and the thread 1 control unit is the next to take control, then an Nth arithmetic unit will execute the instructions of thread 1 based on information contained in advance in the first series of instructions (i.e., the load or jump).

13. Referring to claim 5, Parady has taught a processor control apparatus as described in claim 1. Parady has further taught a second storage element which operates to hold, when one of said arithmetic units executing a first series of instructions from one of said instruction control units is switched to execute a second series of instructions from another instruction control unit, data generated by the second series of instructions under execution by associating the data with that instruction control unit which is executing the second series of instructions. See Fig.3, components 48 and 50. Note that a register file exists for each thread to store data generated during execution of each thread, respectively.

14. Referring to claim 6, Parady has taught a processor control apparatus as described in claim 1. Parady has further taught that it is determined, based on an instruction executing state of each arithmetic unit, one of said arithmetic units to which a new series of instructions is to be issued by one of said instruction control units, and wherein said one instruction control unit is controlled based on the result of the determination so that the new series of instructions are directed to said one arithmetic unit thus determined. More specifically, when the arithmetic units are idle (for instance, in the case of a cache miss), a switch will occur, causing the new series of instructions to be directed to a particular (or more than one) arithmetic unit. For instance, if the thread 1 control unit is the next to issue instructions, and the instructions are all floating point multiplication instructions, then they will all be directed to the FP multiply unit 40 shown in Fig.1.

15. Referring to claim 8, Parady has taught a processor control apparatus as described in claim 1. Parady has further taught that each of said series of instructions includes a series of time sharing instructions for serially driving a plurality of ones of said arithmetic units. In one embodiment of Parady, threads are switched based on a round-robin counter (Fig.3, component 128), which is used to switch between threads at specified time intervals. In essence, the instructions in a series all share an amount of time before that thread (series) is switched out.

16. Claims 1-2, 8-9, 17, 23, 25, and 28 are rejected under 35 U.S.C. 102(e) as being anticipated by Fernando et al., U.S. Patent No. 6,272,616 (herein referred to as Fernando).

17. Referring to claim 1, Fernando has taught a processor control apparatus for controlling a plurality of arithmetic units (Fig.1, components 24 and 26), said processor control apparatus comprising a plurality of instruction control units (Fig.1) issuing a series of instructions to said plurality of arithmetic units, wherein at least one of said instruction control units being operable to switch between a first execution process driving said plurality of arithmetic units by a single series of instructions and a second execution process for driving said plurality of arithmetic units by a plurality of different series of instructions, respectively. See Fig.1 (notice multiplexer 21), and claim 44.

18. Referring to claim 2, Fernando has taught a processor control apparatus as described in claim 1. Fernando has further taught that said at least one instruction control unit each perform a switching process for switching between said first execution process and said second execution process according to information which is contained in advance in a series of instructions. Note that the multiplexer 21 (Fig.1) is what causes a switch in instruction stream execution. This

switch is caused by a change in signal 32i, which is in response to an instruction (CFORK or DFORK) contained in advance in a first instruction stream. See column 5, lines 8-16.

19. Referring to claim 8, Fernando has taught a processor control apparatus as described in claim 1. Fernando has further taught that each of said series of instructions includes a series of time sharing instructions for serially driving a plurality of ones of said arithmetic units. For instance, looking at Fig.7 of Fernando, the series of instructions corresponding to thread 2 execute from time 2 to time 7. Therefore, the thread takes 6 time units to execute and each individual instruction requires some portion of the overall 6 time units for execution. Therefore, each instruction is a time sharing instruction, i.e., each instruction shares the overall 6 time units with the other instructions.

20. Referring to claim 9, Fernando has taught a processor control apparatus as described in claim 1. Fernando has further taught power control elements for controlling power supply to said arithmetic units based on their instruction executing states. See the abstract and column 8, lines 10-12, and note that unused processing elements are deactivated in order to conserve power.

21. Referring to claim 17, Fernando has taught a processor control apparatus comprising:

a) a single instruction memory for storing a plurality of series of instructions to be executed by a plurality of arithmetic units. See Fig.1, components 12 and 26. Fig.7 shows multiple series of instructions being executed by different units.

b) an instruction decoder for decoding a series of instructions from said instruction memory, and outputting a decoded result to any of said plurality of arithmetic units. See Fig.1, component 22a and 22b.

c) a selector selectively switching between a plurality of series of instructions from said instruction memory to be decoded by said instruction decoder, and supplying a series of instructions thus selected to said instruction decoder. See Fig.1, component 21, and claim 44 of Fernando.

d) wherein said instruction memory has a plurality of ports for issuing said series of instructions to said instruction decoder. See Fig.1 component 12 and note that multiple ports are connected to multiple fetch units and ultimately to multiple decoders.

22. Referring to claim 23, Fernando has taught a processor comprising:

a) a plurality of arithmetic units. See Fig.1, components 24a, 24b, and 26.

b) a plurality of instruction control units for issuing a series of instructions to drive said arithmetic units in a controlled manner. See Fig.1 and note a first instruction control unit would comprise at least components 20a, 22a, and 12, whereas a second unit would comprise at least components 20b, 22b, and 12. Each of these units issues instructions to the arithmetic units.

c) wherein some of said instruction control units are operable to switch between a first execution process for driving said plurality of arithmetic units by a single series of instructions and a second execution process for driving said plurality of arithmetic units by a plurality of different series of instructions, respectively. See Fig.1, component 21 and claim 44. Note that the MUX can either select a first series of instruction from bus 36 or a second series from bus 35.

23. Referring to claim 25, Fernando has taught a processor comprising:

a) a plurality of arithmetic units. See Fig.1, components 24 and 26.

b) a single instruction memory for storing a plurality of series of instructions to be executed by a plurality of arithmetic units. See Fig.1, components 12 and 26. Fig.7 shows multiple series of instructions being executed by different units.

c) an instruction decoder for decoding a series of instructions from said instruction memory, and outputting a decoded result to any of said plurality of arithmetic units. See Fig.1, component 22a and 22b.

d) a selector for selectively switching between a plurality of series of instructions from said instruction memory to be decoded by said instruction decoder, and supplying a series of instructions thus selected to said instruction decoder. See Fig.1, component 21, and claim 44 of Fernando.

e) wherein said instruction memory has a plurality of ports for issuing said series of instructions to said instruction decoder. See Fig.1 component 12 and note that multiple ports are connected to multiple fetch units and ultimately to multiple decoders.

24. Referring to claim 28, Fernando has taught a processor controlling method usable with a plurality of instruction control units for controlling a plurality of arithmetic units to execute a plurality of series of instructions, said method comprising:

a) prescribing, in advance in a series of instructions which is to be performed, synchronous execution in which a plurality of predetermined ones of said arithmetic units are synchronously driven by a single series of instructions, or independent execution in which the plurality of predetermined arithmetic units are independently driven by a plurality of respective series of instructions. A single series of instructions may be used to drive a plurality of arithmetic units. This would occur when the selector 21 shown in Fig.1 selects the instructions from bus 36. This

corresponds to Fig.7 at times 10-12 when a single series of instruction (thread 1) is executed by two different data paths (arithmetic units), and hence results in the "combined control" illustrated in the figure. On the other hand, independent execution of a plurality of series of instructions by a plurality of arithmetic units is also shown in Fig.7. For instance, two series (corresponding to thread 1 and 2) are independently executed by data paths 1 and 2 (arithmetic units). This would occur when MUX 21 in Fig.1 selects the group of instructions from bus 35.

b) switching between the predetermined arithmetic units for performing a series of instructions based on the contents of the prescription therein. Clearly from the examples given in Fig.7, synchronous execution of a single series or independent execution of multiple series of instructions is switched back and forth. See Fig.7, and notice data paths 1 and 2 from time 1-12.

Claim Rejections - 35 USC § 103

25. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

26. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Parady, as applied above, in view of Dowling, U.S. Patent No. 6,170,051.

27. Referring to claim 7, Parady has taught a processor control apparatus as described in claim 1. Parady has not taught that each of said series of instructions includes a VLIW type instruction. However, Dowling has taught series of instructions that comprise VLIW instructions. See the abstract. As is known in the art, and explained in column 1, lines 18-29,

and lines 45-51, of Dowling, VLIW instructions allow for the utilization of multiple functional units in the same cycle, as opposed to just one per cycle. This (instruction level parallelism), in turn, increases throughput. One of ordinary skill in the art would have recognized that VLIW instructions would be beneficial and implementable in Parady because Parady has taught multiple functional units, which when used in conjunction with VLIW instructions, would be utilized more efficiently in order to increase throughput. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady to include VLIW instructions.

28. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Parady, as applied above, in view of Fernando, as applied above.

29. Referring to claim 9, Parady has taught a processor control apparatus as described in claim 1. Parady has not taught power control elements for controlling power supply to said arithmetic units based on their instruction executing states. However, Fernando has taught such a concept. See the abstract and column 8, lines 10-12. Note that when a particular functional unit is not in use, it will be powered down, thereby reducing the overall power consumption of the system. As is seen in Fig.1 of Parady, many functional units exist which can execute instructions. A person of ordinary skill in the art would have recognized that by implementing the power-saving concept of Fernando into the system of Parady, power consumption would be reduced by powering down those functional units which are not in use. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady to include the powering mechanism taught by Fernando.

30. Claims 10-12, 15-16, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fernando, as applied above.

31. Referring to claim 10, Fernando has taught a processor control apparatus comprising:

a) a single instruction memory for storing a plurality of series of instructions and supplying them to arithmetic units. See Fig.1, components 12 and 26. Fernando has not taught a plurality of instruction memories for storing a plurality of series of instructions to be executed by a plurality of arithmetic units. However, Official Notice is taken that having a plurality of memories for storing independent series of instructions is well known and accepted in the art. In addition, as shown in Nerwin v. Erlichman, 168 USPQ 177 (1969), to make separable is generally not given patentable weight or would have been an obvious improvement. For instance, a person of ordinary skill in the art would have recognized that by implementing a plurality of memories instead of a single memory with concurrent access capabilities, as taught by Fernando (evident in Fig.7 where multiple streams are fetched at once), then the circuitry required to allow for concurrent access would be eliminated, thereby reducing the complexity and cost of the system. Consequently, it would have been obvious to replace Fernando's single instruction memory with a plurality of instruction memories.

b) an instruction decoder for decoding a series of instructions from said instruction memories, and outputting a decoded result to any of said plurality of arithmetic units. See Fig.1, component 22a, 22b, and 26.

c) and a selector for selectively switching between a plurality of series of instructions from said instruction memories to be decoded by said instruction decoder, and supplying a series of

instructions thus selected to said instruction decoder. See Fig.1, component 21, and claim 44 of Fernando.

32. Referring to claim 11, Fernando has taught a processor control apparatus as described in claim 10. Fernando has further taught that some of said plurality of series of instructions contain information about selective switching between said series of instructions to be performed by said selector, and wherein said instruction decoder decodes said information contained in a series of instructions, and outputs a switching instruction to said selector. See column 6, lines 41-54, and column 7, lines 17-29, and note that when a CFORK or DFORK is detected by decoder 22a, a signal 32i (Fig.1) is generated, which then causes the multiplexer 21 (Fig.1) to select another series of instructions, which will then be decoded.

33. Referring to claim 12, Fernando has taught a processor control apparatus as described in claim 10. Fernando has further taught that some of said plurality of series of instructions contain a synchronizing instruction for allowing a first predetermined one of said arithmetic units and a second predetermined arithmetic unit to synchronously perform processes, and wherein when said synchronizing instruction is issued to said first predetermined arithmetic unit, said first predetermined arithmetic unit is set in a wait state, and an instruction decoder of said second predetermined arithmetic unit does not output a switching instruction to its associated selector if a process is being executed by said second predetermined arithmetic unit upon issuance of said synchronizing instruction, and does not release the wait state of said first predetermined arithmetic unit until said second predetermined arithmetic unit completes said process. See column 7, lines 52-67. Note that a WAIT instruction will pause the first arithmetic unit until a DJOIN instruction is executed by the second arithmetic unit (synchronization). Furthermore, the

stream will not be switched as the second arithmetic unit will continue executing the same thread until it is finished (upon a DJOIN instruction). See Fig.7.

34. Referring to claim 15, Fernando has taught a processor control apparatus as described in claim 10. Fernando has further taught that each of said series of instructions includes a series of time sharing instructions for serially driving a plurality of ones of said arithmetic units. For instance, looking at Fig.7 of Fernando, the series of instructions corresponding to thread 2 execute from time 2 to time 7. Therefore, the thread takes 6 time units to execute and each individual instruction requires some portion of the overall 6 time units for execution. Therefore, each instruction is a time sharing instruction, i.e., each instruction shares the overall 6 time units with the other instructions.

35. Referring to claim 16, Fernando has taught a processor control apparatus as described in claim 10. Fernando has further taught power control elements for controlling power supply to said arithmetic units based on their instruction executing states. See the abstract and column 8, lines 10-12, and note that unused processing elements are deactivated in order to conserve power.

36. Referring to claim 24, Fernando has taught a processor comprising:

a) a plurality of arithmetic units. See Fig.1, components 24 and 26.

b) a single instruction memory for storing a plurality of series of instructions and supplying them to arithmetic units. See Fig.1, components 12. Fernando has not taught a plurality of instruction memories for storing a plurality of series of instructions to be executed by a plurality of arithmetic units. However, Official Notice is taken that having a plurality of memories for storing independent series of instructions is well known and accepted in the art. In addition, as

shown in Nerwin v. Erlichman, 168 USPQ 177 (1969), to make separable is generally not given patentable weight or would have been an obvious improvement. For instance, a person of ordinary skill in the art would have recognized that by implementing a plurality of memories instead of a single memory with concurrent access capabilities, as taught by Fernando (evident in Fig.7 where multiple streams are fetched at once), then the circuitry required to allow for concurrent access would be eliminated, thereby reducing the complexity and cost of the system. Consequently, it would have been obvious to replace Fernando's single instruction memory with a plurality of instruction memories.

c) an instruction decoder for decoding a series of instructions from said instruction memories, and outputting a decoded result to any of said plurality of arithmetic units. See Fig.1, component 22a and 22b.

d) and a selector for selectively switching between a plurality of series of instructions from said instruction memories to be decoded by said instruction decoder, and supplying a series of instructions thus selected to said instruction decoder. See Fig.1, component 21, and claim 44 of Fernando. Note that the MUX can either select a first series of instruction from bus 36 or a second series from bus 35.

37. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Fernando, as applied above, in view of Parady, as applied above.

38. Referring to claim 13, Fernando has taught a processor control apparatus as described in claim 10.

a) Fernando has not explicitly taught an instruction queue for temporarily storing, at a stage prior to said selector, a series of instructions to be transmitted from a second one of said instruction memories different from a first one of said instruction memories which stores a series of instructions being executed by said first predetermined arithmetic unit. However, Parady has taught employing multiple instruction queues for holding different series of instructions before one of them is selected. See Fig.3, components 102-108. A person of ordinary skill in the art would have recognized that an instruction queue allows a system to hold a series of instructions within the system. Instructions from the queue will be retrieved faster than if they had to be retrieved from main memory since main memory is slower than an on-chip memory, such as Parady's queues. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando to include instruction queues on-chip to hold series of instruction so that the instructions are retrieved faster than they would from main memory.
b) and a determiner for determining, based on a series of instructions being executed, whether or not the process being performed by said first predetermined arithmetic unit can be interrupted, said determiner operating to output, if the process can be interrupted, an interrupt signal for interrupting the issuance of the series of instructions to said first instruction memory which is a source of the series of instructions being executed, and generate a switching instruction to said selector to switch to a series of instructions from said instruction queue. Looking at Fig.1, it should be realized that if the first arithmetic unit 24b is executing a stream of instructions from bus 36 and a DFORK instruction is encountered by decoder 22a, then the MUX for the first arithmetic unit will be configured such that the execution of a first stream of instructions is interrupted in order to execute a second stream of instructions from bus 35. See column 5, lines

17-25, and column 7, lines 17-29. So, in essence, Fernando has taught interrupting a first process in order to execute a second process.

39. Claims 14, 18-22, and 26-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fernando, as applied above, in view of Dowling, as applied above.

40. Referring to claim 14, Fernando has taught a processor control apparatus as described in claim 10. Fernando has not taught that each of said series of instructions includes a VLIW type instruction. However, Dowling has taught series of instructions that comprise VLIW instructions. See the abstract. As is known in the art, and explained in column 1, lines 18-29, and lines 45-51, of Dowling, VLIW instructions allow for the utilization of multiple functional units in the same cycle, as opposed to just one per cycle. This (instruction level parallelism), in turn, increases throughput. One of ordinary skill in the art would have recognized that VLIW instructions would be beneficial and implementable in Fernando because Fernando has taught multiple functional units (Fig.1, components 26), which when used in conjunction with VLIW instructions, would be utilized more efficiently in order to increase throughput. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando to include VLIW instructions.

41. Referring to claim 18, Fernando has taught a processor control apparatus for controlling a plurality of arithmetic units (Fig.1, components 24a and 24b), said processor control apparatus comprising a plurality of instruction control units for instructing said arithmetic units to execute a series of instructions, wherein each of said instruction control units includes:

a) a single instruction memory for storing a plurality of series of instructions and supplying them to arithmetic units. See Fig.1, components 12 and 26. Fernando has not taught that each instruction control unit includes an instruction memory for storing a plurality of series of instructions, i.e. a plurality of instruction memories. However, Official Notice is taken that having a plurality of memories for storing series of instructions is well known and accepted in the art. In addition, as shown in Nerwin v. Erlichman, 168 USPQ 177 (1969), to make separable is generally not given patentable weight or would have been an obvious improvement. For instance, a person of ordinary skill in the art would have recognized that by implementing a plurality of memories instead of a single memory with concurrent access capabilities, as taught by Fernando (evident in Fig.7 where multiple streams are fetched at once), then the circuitry required to allow for concurrent access would be eliminated, thereby reducing the complexity and cost of the system. Consequently, it would have been obvious to replace Fernando's single instruction memory with a plurality of instruction memories.

b) an instruction decoder for decoding a series of instructions and supplying the decoded series of instructions to an associated one of said arithmetic units. See Fig.1, components 22a and 22b (note that the control units may each comprise at least a decoder, fetch unit, and instruction memory).

c) wherein some of said instruction control units each have an instruction control selector for selectively switching between a first series of instructions from a first instruction memory of one of said instruction control units and a second series of instructions from a second instruction memory of another instruction control unit different from said one instruction control unit to output one of said first and second series of instructions thus selected to said instruction decoder.

See Fig.1, component 21 and note that either a first series of instructions along bus 36 may be selected for decoding or a second series of instructions from bus 35 may be selected for decoding.

d) wherein each of said arithmetic units includes a register file (say 25b) for storing data during execution and the ability to access the register file of another arithmetic unit (25a) via bus 38 by using MOVE instructions. See column 6, lines 16-32. Fernando has not taught that each arithmetic unit includes a first register file and a second register file for storing data generated by said first and second series of instructions, respectively, which are supplied from said first and second instruction memories and decoded by an instruction decoder of an associated one of said instruction control units, and an arithmetic unit selector for selectively switching between said data generated by said first and second series of instructions being executed and stored in said first and second register files, respectively, according to an instruction from said associated instruction decoder to supply a selected one of said first and second series of instructions to a calculator.. However, Dowling has taught such a concept. Note from Fig.2, that arithmetic unit 200 includes two register sets 160 and 260 for use in executing two different instruction streams 130 and 230. Based on the stream, one of the register sets is selected by selector 290. By adding a second set of registers to the arithmetic unit, according to column 9, lines 59-62, the processing hardware can quickly switch both internal context and external context, i.e. quickly switch execution from one stream of instructions to another, as Fernando has taught. A person of ordinary skill in the art would have also recognized that by having a second register set in an arithmetic unit, the MOVE instruction would not be required to move data from one set to another in order to facilitate execution of another stream. This would also reduce the complexity

of the system. Therefore, in order to speed up the system and reduce complexity, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando such that each arithmetic unit includes a second register set.

42. Referring to claim 19, Fernando in view of Dowling has taught a processor control apparatus as described in claim 18. Fernando has not taught that each of said series of instructions includes a VLIW type instruction. However, Dowling has taught series of instructions that comprise VLIW instructions. See the abstract. As is known in the art, and explained in column 1, lines 18-29, and lines 45-51, of Dowling, VLIW instructions allow for the utilization of multiple functional units in the same cycle, as opposed to just one per cycle. This (instruction level parallelism), in turn, increases throughput. One of ordinary skill in the art would have recognized that VLIW instructions would be beneficial and implementable in Fernando because Fernando has taught multiple functional units, which when used in conjunction with VLIW instructions, would be utilized more efficiently in order to increase throughput. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando to include VLIW instructions.

43. Referring to claim 20, Fernando in view of Dowling has taught a processor control apparatus as described in claim 18. Fernando has further taught that each of said series of instructions includes a series of time sharing instructions for serially driving a plurality of ones of said arithmetic units. For instance, looking at Fig.7 of Fernando, the series of instructions corresponding to thread 2 execute from time 2 to time 7. Therefore, the thread takes 6 time units to execute and each individual instruction requires some portion of the overall 6 time units for

execution. Therefore, each instruction is a time sharing instruction, i.e., each instruction shares the overall 6 time units with the other instructions.

44. Referring to claim 21, Fernando in view of Dowling has taught a processor control apparatus as described in claim 18. Fernando has further taught power control elements for controlling power supply to said arithmetic units based on their instruction executing states. See the abstract and column 8, lines 10-12, and note that unused processing elements are deactivated in order to conserve power.

45. Referring to claim 22, Fernando has taught a processor control apparatus for controlling a plurality of arithmetic units (Fig.1, components 24 and 26), said processor control apparatus comprising a plurality of instruction control units (Fig.1) for instructing said arithmetic units to execute a series of instructions.

a) wherein said instruction control units have a single instruction memory used in common for storing a plurality of series of instructions (see Fig.1, component 12), and each includes an instruction decoder for decoding a series of instructions and supplying the decoded series of instructions to an associated one of said arithmetic units (see Fig.1, components 22a and 22b, and note that the control units may each comprise at least a decoder, fetch unit, and instruction memory), and said single instruction memory has a plurality of ports for issuing said series of instructions to said respective instruction decoders (See Fig.1 component 12 and note that multiple ports are connected to multiple fetch units and ultimately to multiple decoders).

b) wherein some of said instruction control units each have an instruction control selector for selectively switching between a first series of instructions from a first instruction memory of one of said instruction control units and a second series of instructions from a second instruction

memory of another instruction control unit different from said one instruction control unit to output one of said first and second series of instructions thus selected to said instruction decoder. See Fig.1, component 21 and note that either a first series of instructions along bus 36 may be selected for decoding or a second series of instructions from bus 35 may be selected for decoding.

c) wherein each of said arithmetic units includes a register file (say 25b) for storing data during execution and the ability to access the register file of another arithmetic unit (25a) via bus 38 by using MOVE instructions. See column 6, lines 16-32. Fernando has not taught that each arithmetic unit includes a first register file and a second register file for storing data generated by said first and second series of instructions, respectively, which are supplied from said first and second instruction memories and decoded by an instruction decoder of an associated one of said instruction control units, and an arithmetic unit selector for selectively switching between said data generated by said first and second series of instructions being executed and stored in said first and second register files, respectively, according to an instruction from said associated instruction decoder to supply a selected one of said first and second series of instructions to a calculator.. However, Dowling has taught such a concept. Note from Fig.2, that arithmetic unit 200 includes two register sets 160 and 260 for use in executing two different instruction streams 130 and 230. Based on the stream, one of the register sets is selected by selector 290. By adding a second set of registers to the arithmetic unit, according to column 9, lines 59-62, the processing hardware can quickly switch both internal context and external context, i.e. quickly switch execution from one stream of instructions to another, as Fernando has taught. A person of ordinary skill in the art would have also recognized that by having a second register set in an

arithmetic unit, the MOVE instruction would not be required to move data from one set to another in order to facilitate execution of another stream. This would also reduce the complexity of the system. Therefore, in order to speed up the system and reduce complexity, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando such that each arithmetic unit includes a second register set.

46. Referring to claim 26, Fernando has taught a processor comprising:

a) a plurality of arithmetic units (Fig.1, components 24 and 26).

b) a plurality of instruction control units (Fig.1) for driving said arithmetic units in a controlled manner, wherein each of said instruction control units includes:

c) a single instruction memory for storing a plurality of series of instructions and supplying them to arithmetic units. See Fig.1, components 12 and 26. Fernando has not taught that each instruction control unit includes an instruction memory for storing a plurality of series of instructions, i.e. a plurality of instruction memories. However, Official Notice is taken that having a plurality of memories for storing series of instructions is well known and accepted in the art. In addition, as shown in Nerwin v. Erlichman, 168 USPQ 177 (1969), to make separable is generally not given patentable weight or would have been an obvious improvement. For instance, a person of ordinary skill in the art would have recognized that by implementing a plurality of memories instead of a single memory with concurrent access capabilities, as taught by Fernando (evident in Fig.7 where multiple streams are fetched at once), then the circuitry required to allow for concurrent access would be eliminated, thereby reducing the complexity and cost of the system. Consequently, it would have been obvious to replace Fernando's single instruction memory with a plurality of instruction memories.

d) an instruction decoder for decoding a series of instructions and supplying the decoded series of instructions to an associated one of said arithmetic units. See Fig.1, components 22a and 22b (note that the control units may each comprise at least a decoder, fetch unit, and instruction memory).

e) wherein some of said instruction control units each have an instruction control selector for selectively switching between a first series of instructions from a first instruction memory of one of said instruction control units and a second series of instructions from a second instruction memory of another instruction control unit different from said one instruction control unit to output one of said first and second series of instructions thus selected to said instruction decoder. See Fig.1, component 21 and note that either a first series of instructions along bus 36 may be selected for decoding or a second series of instructions from bus 35 may be selected for decoding.

f) wherein each of said arithmetic units includes a register file (say 25b) for storing data during execution and the ability to access the register file of another arithmetic unit (25a) via bus 38 by using MOVE instructions. See column 6, lines 16-32. Fernando has not taught that each arithmetic unit includes a first register file and a second register file for storing data generated by said first and second series of instructions, respectively, which are supplied from said first and second instruction memories and decoded by an instruction decoder of an associated one of said instruction control units, and an arithmetic unit selector for selectively switching between said data generated by said first and second series of instructions being executed and stored in said first and second register files, respectively, according to an instruction from said associated instruction decoder to supply a selected one of said first and second series of instructions to a

calculator.. However, Dowling has taught such a concept. Note from Fig.2, that arithmetic unit 200 includes two register sets 160 and 260 for use in executing two different instruction streams 130 and 230. Based on the stream, one of the register sets is selected by selector 290. By adding a second set of registers to the arithmetic unit, according to column 9, lines 59-62, the processing hardware can quickly switch both internal context and external context, i.e. quickly switch execution from one stream of instructions to another, as Fernando has taught. A person of ordinary skill in the art would have also recognized that by having a second register set in an arithmetic unit, the MOVE instruction would not be required to move data from one set to another in order to facilitate execution of another stream. This would also reduce the complexity of the system. Therefore, in order to speed up the system and reduce complexity, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando such that each arithmetic unit includes a second register set.

47. Referring to claim 27, Fernando has taught a processor comprising:

a) a plurality of arithmetic units (Fig.1, components 24 and 26).

b) a plurality of instruction control units (Fig.1) for driving said arithmetic units in a controlled manner.

c) wherein said instruction control units have a single instruction memory used in common for storing a plurality of series of instructions (see Fig.1, component 12), and each includes an instruction decoder for decoding a series of instructions and supplying the decoded series of instructions to an associated one of said arithmetic units (see Fig.1, components 22a and 22b, and note that the control units may each comprise at least a decoder, fetch unit, and instruction memory), and said single instruction memory has a plurality of ports for issuing said series of

instructions to said respective instruction decoders (See Fig.1 component 12 and note that multiple ports are connected to multiple fetch units and ultimately to multiple decoders).

e) wherein some of said instruction control units each have an instruction control selector for selectively switching between a first series of instructions from a first instruction memory of one of said instruction control units and a second series of instructions from a second instruction memory of another instruction control unit different from said one instruction control unit to output one of said first and second series of instructions thus selected to said instruction decoder. See Fig.1, component 21 and note that either a first series of instructions along bus 36 may be selected for decoding or a second series of instructions from bus 35 may be selected for decoding.

f) wherein each of said arithmetic units includes a register file (say 25b) for storing data during execution and the ability to access the register file of another arithmetic unit (25a) via bus 38 by using MOVE instructions. See column 6, lines 16-32. Fernando has not taught that each arithmetic unit includes a first register file and a second register file for storing data generated by said first and second series of instructions, respectively, which are supplied from said first and second instruction memories and decoded by an instruction decoder of an associated one of said instruction control units, and an arithmetic unit selector for selectively switching between said data generated by said first and second series of instructions being executed and stored in said first and second register files, respectively, according to an instruction from said associated instruction decoder to supply a selected one of said first and second series of instructions to a calculator.. However, Dowling has taught such a concept. Note from Fig.2, that arithmetic unit 200 includes two register sets 160 and 260 for use in executing two different instruction streams

130 and 230. Based on the stream, one of the register sets is selected by selector 290. By adding a second set of registers to the arithmetic unit, according to column 9, lines 59-62, the processing hardware can quickly switch both internal context and external context, i.e. quickly switch execution from one stream of instructions to another, as Fernando has taught. A person of ordinary skill in the art would have also recognized that by having a second register set in an arithmetic unit, the MOVE instruction would not be required to move data from one set to another in order to facilitate execution of another stream. This would also reduce the complexity of the system. Therefore, in order to speed up the system and reduce complexity, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Fernando such that each arithmetic unit includes a second register set.

Response to Arguments

48. Applicant's arguments filed on October 18, 2004, have been fully considered but they are not persuasive.

49. Applicant argues the novelty/rejection of claim 1 on page 16 of the remarks, in substance that:

> "The Examiner contends that these features are described by Parady in that components 150 and 154 teach control units for issuing a series of instructions to said plurality of arithmetic units. Applicants submit however that elements 154 are four instruction buffers. (See, Parady, col. 5, line 8). Nothing in Parady discusses that such an instruction buffer is operable to switch between a first execution process driving said plurality of arithmetic units by a single series of instructions and a second execution process driving the arithmetic unit."

50. These arguments are not found persuasive for the following reasons:

a) In the rejection of claim 1, the examiner specified that components 154 and 150 made up the instruction control unit. The same components are shown in Fig.3 only renumbered (not the

dispatch unit and the instruction buffers). The examiner asserts that one of these buffers is selected to provide instructions to the dispatch unit, from which they will be sent to execution units. As the abstract of Parady states, when a long latency event occurs, a thread switch occurs, and a new instruction buffer begins to supply instructions to the dispatch unit. This is controlled by a signal from the thread switching logic. Consequently, these components (buffers and dispatcher) do in fact switch between a first execution process and a second execution process. The first execution process would be when thread 1 is executing (for example), and after a thread switch, thread 2 would be a second execution process (for example).

51. Applicant argues the novelty/rejection of claim 1 on page 17 of the remarks, in substance that:

> "In contrast to the cited art, independent claim 1 recites a processor control apparatus "wherein at least one of said instruction control units being operable to switch between a first execution process driving said plurality of arithmetic units by a single series of instructions and a second execution process driving said plurality of arithmetic units by a plurality of different series of instructions, respectively." Thus, according to aspects of the present invention not only scalar commands (SINGLE), but also VLIM commands (of course, SIMD-like operation is possible) and MIMD commands can be performed. Applicants submit that Fernando does not teach these features, for example, Fernando does not teach that operations performed by VLIW commands or the like can be provided. Fernando merely proposes a method to switch between a scalar command (SINGLE), an SIMD command, and an MIMD command. For example, Fernando's Fig. 5 shows that upon execution of MIMD commands, fetching and arithmetic execution of a SIDE A and a SIDE B are dependent from each other."
>
> "Applicants further submit that Fernando does not discuss instruction control units being operable to switch between a first execution process driving said plurality of arithmetic units by a single series of instructions. Instead, Fernando merely teaches (col. 4, starting at line 20) "processors having any number of parallel instruction pipelines.'"

52. These arguments are not found persuasive for the following reasons:

a) In response to applicant's first argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., switching between VLIM and scalar commands) are not recited in claims 1-2 and 8-9. Although the claims

are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). All applicant is claiming is switching between a single series of instructions and a plurality of different series of instructions. Looking at Fig.1, functional units 26 may execute a single series of instructions from either path 35 or 36 or a different series of instructions from the other path of paths 35 and 36 (based on the selector 21). A series of instructions is nothing more than some number of instructions. And a program may contain a plurality of series of instructions. For instance, if a program coming along path 35 comprises six instructions, it could be viewed as two series of three instructions.

b) It is not clear what the applicant is arguing. Clearly, from Fig.1 of Fernando, a selector 21 may be used to switch between execution processes which drive the arithmetic units. Either a first process (single series of instructions) will be selected from one of paths 35 and 36 or a different series will be selected from the other of paths 35 and 36.

53. Applicant argues the novelty/rejection of claim 7 on page 18 of the remarks, in substance that:

> "As provided in MPEP 2143 entitled Basic Requirements of a Prima Facie Case of Obviousness: The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Applicants submit there is no motivation to combine the references in an manner as suggested by the Examiner."

54. These arguments are not found persuasive for the following reasons:

a) The test of obviousness is:

> "whether the teachings of the prior art, taken as a whole, would have made obvious the claimed invention," In re Gorman, 933 F.2d at 986, 18 USPQ2d at 1888.

> Subject matter is unpatentable under section 103 if it "'would have been obvious . . . to a person having ordinary skill in the art.' **While there must be some teaching, reason, suggestion, or motivation to combine existing elements to produce the claimed device, it is not necessary that the cited references or prior art specifically suggest making the combination.**" In re Nilssen, 851 F.2d 1401, 1403, 7 USPQ2d 1500, 1502 (Fed. Cir. 1988).
>
> "Such suggestion or motivation to combine prior art teachings can derive solely from the existence of a teaching, which one of ordinary skill in the art would be presumed to know, and the use of that teaching to solve the same [or] similar problem which it addresses." In re Wood, 599 F.2d 1032, 1037, 202 USPQ 171, 174 (CCPA 1979).
>
> "In sum, it is off the mark for litigants to argue, as many do, that an invention cannot be held to have been obvious unless a suggestion to combine prior art teachings is found in a specific reference."
>
> Entire quote from In re Oetiker, 24 USPQ2d 1443 (CAFC 1992)."
>
> Accordingly, Parady and Dowling are not required to disclose or specifically suggest particular elements. Instead the measure is what the teachings of Parady and Dowling would suggest to one of ordinary skill in the art, not what Parady and Dowling specifically suggests.

From this, it can be seen that the prior art does not specifically have to state a reason to combine. Instead, the reason could be obvious to one of ordinary skill in the art. In this situation, the examiner had given a valid motivation in the original rejection of claim 7. To repeat, VLIW instructions allow for the utilization of multiple functional units in the same cycle, as opposed to just one per cycle. This (instruction level parallelism), in turn, increases throughput. One of ordinary skill in the art would have recognized that VLIW instructions would be beneficial and implementable in Parady because Parady has taught multiple functional units, which when used in conjunction with VLIW instructions, would be utilized more efficiently in order to increase throughput.

55. Applicant argues the novelty/rejection of claim 9 on page 19 of the remarks, in substance that:

> "Applicants submit that Fernando does not teach controlling power supply to arithmetic units based on their instruction execution states. Fernando merely discusses col. 8, lines 10-12 'deactivating processor elements in secondary pipelines which are not needed.'"

56. These arguments are not found persuasive for the following reasons:

a) Deactivating arithmetic units which are not needed is the same as deactivating arithmetic units which based on execution state. For instance, if they are not needed, then it could be said that they are in a state of "non-execution," and consequently, disabled to save power. On the other hand, units which are needed could be though of as being in an "execution" state, and therefore powered up.

57. Applicant argues the novelty/rejection of claim 10 on page 19 of the remarks, in substance that:

> "Nevertheless, the Examiner rejects independent claim 10 taking Official Notice that having a plurality of memories for storing independent series of instructions is well known. Since the taking of Official Notice is unsupported, the rejection should be withdrawn and the claims allowed."

58. These arguments are not found persuasive for the following reasons:

a) From MPEP 2144.03[R-1], "If applicant adequately traverses the examiner's assertion of official notice, the examiner must provide documentary evidence in the next Office action if the rejection is to be maintained. See 37 CFR 1.104(c)(2). See also Zurko, 258 F.3d at 1386, 59 USPQ2d at 1697." As a result, the examiner would like to direct applicant's attention to column 4, line 66, to column 5, line 10, of Ramagopal et al., U.S. Patent No. 6,606,684 (hereafter Ramagopal). Ramagopal has taught that multiple memory banks are advantageous because

parallel access may be achieved with minimal hardware overhead (less complex circuitry) and cost. As a result, the examiner feels that the Official Notice is properly supported. In addition, even if the Official Notice were not properly supported, applicant has not addressed the case law brought to applicant's attention which states that it would be obvious to separate a component. The examiner encourages applicant to review the rejection of claim 10.

59. Applicant argues the novelty/rejection of claims 23, 25, and 28 on page 20 of the remarks, in substance that:

> "Applicants submit that Fernando does not discuss instruction control units being operable to switch between a first execution process driving said plurality of arithmetic units by a single series of instructions. Instead, Fernando merely teaches (col. 4, starting at line 20) "processors having any number of parallel instruction pipelines.'"

60. These arguments are not found persuasive for the following reasons:

a) It is not clear what the applicant is arguing. Clearly, from Fig.1 of Fernando, a selector 21 may be used to switch between execution processes which drive the arithmetic units. Either a first process (single series of instructions) will be selected from one of paths 35 and 36 or a different series will be selected from the other of paths 35 and 36.

61. Applicant argues the novelty/rejection of claim 13 on page 21 of the remarks, in substance that:

> "Applicants submit there is no motivation to combine the references in a manner as suggested by the Examiner. Since Fernando discusses a pipeline including fetch stages to retrieve instructions, Applicants submit there is not a motivation to modify Fernando as the Examiner contends. Since there is no motivation to combine the cited ad, the rejection should be withdrawn and claim 13 allowed."

62. These arguments are not found persuasive for the following reasons:

a) As described above, the prior art does not specifically have to state a reason to combine. Instead, the reason could be provided by one of ordinary skill in the art. In this situation, the examiner had given a valid motivation in the original rejection. To repeat, an instruction queue(s) allows a system to hold a series of instructions within the system. Instructions from the queue will be retrieved faster than if they had to be retrieved from main memory since main memory is slower than an on-chip memory. Even if Fernando employs fetch stages to retrieve instructions, Fernando would benefit from these instruction queues, as instead of fetching from main memory, which is slow, fetching would occur from the instruction queues (which hold instructions already brought in from memory).

63. Applicant argues the novelty/rejection of claims 14 and 19 on page 21 of the remarks, in substance that:

> "As provided in MPEP 52143 entitled Basic Requirements of a Prima Facie Case of Obviousness: (the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Applicants submit there is no motivation to combine the references in an manner as suggested by the Examiner. Since there is no motivation to combine the cited art, the rejection should be withdrawn and claims 14 and 19 allowed."

64. These arguments are not found persuasive for the following reasons:

a) As described above, the prior art does not specifically have to state a reason to combine. Instead, the reason could have been obvious to and then provided by one of ordinary skill in the art. In this situation, the examiner had given a valid motivation in the original rejection. To repeat, VLIW instructions allow for the utilization of multiple functional units in the same cycle, as opposed to just one per cycle. This (instruction level parallelism), in turn, increases throughput. One of ordinary skill in the art would have recognized that VLIW instructions

would be beneficial and implementable in Fernando because Fernando has taught multiple functional units, which when used in conjunction with VLIW instructions, would be utilized more efficiently in order to increase throughput.

65. Applicant argues the novelty/rejection of claims 18 and 20-21 on page 21 of the remarks, in substance that:

> "Nevertheless, the Examiner rejects claim 18 taking Official Notice that having a plurality of memories for storing independent series of instructions is well known. Since the examiner's contentions are unsupported, the rejection should be withdrawn and the claims allowed."

66. These arguments are not found persuasive for the following reasons:

a) From MPEP 2144.03[R-1], "If applicant adequately traverses the examiner's assertion of official notice, the examiner must provide documentary evidence in the next Office action if the rejection is to be maintained. See 37 CFR 1.104(c)(2). See also Zurko, 258 F.3d at 1386, 59 USPQ2d at 1697." As a result, the examiner would like to direct applicant's attention to column 4, line 66, to column 5, line 10, of Ramagopal et al., U.S. Patent No. 6,606,684 (hereafter Ramagopal). Ramagopal has taught that multiple memory banks are advantageous because parallel access may be achieved with minimal hardware overhead (less complex circuitry) and cost. As a result, the examiner feels that the Official Notice is properly supported. In addition, even if the Official Notice were not properly supported, applicant has not addressed the case law brought to applicant's attention which states that it would be obvious to separate a component. The examiner encourages applicant to review the rejection of claim 18.

67. Applicant argues the novelty/rejection of claims 22 and 26-27 on page 22 of the remarks, in substance that:

> "The Examiner contends that there is motivation to modify Fernando in view of Dowling to reduce complexity. As provided in MPEP 52143 entitled Basic Requirements of a Prima Facie Case of Obviousness: (the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). Applicants submit there is no motivation to combine the references in an manner as suggested by the Examiner and that such a combination would increase complexity."

68. These arguments are not found persuasive for the following reasons:

a) As described above, the prior art does not specifically have to state a reason to combine. However, in this case, Dowling has specifically stated the motivation. To repeat, by adding a second set of registers to the arithmetic unit, according to column 9, lines 59-62, the processing hardware can quickly switch both internal context and external context, i.e. quickly switch execution from one stream of instructions to another, as Fernando has taught. In addition, the examiner gave additional motivation, the motivation being that by having a second register set in an arithmetic unit, the MOVE instruction would not be required to move data from one set to another in order to facilitate execution of another stream. This would also reduce the complexity of the system. That is, when instruction streams are switched, the system would no longer have to move data corresponding to the old stream out to a backup store and the data corresponding to the new stream into the single register file. It is in this manner that complexity would be reduced. Also, even if complexity were increased, as applicant contends, the system speed will still be increased, as the examiner had stated (this is due to Dowling's motivation). And, any system will have advantages and disadvantages and simply because a disadvantage might exist (complexity) does not mean that one wouldn't be motivated to make a combination to take advantage of a different aspect of the combination (speed).

69. Finally, the examiner would like to bring to applicant's attention a mistake made in the first action. Claims 17, 23, 25, and 28 were initially and mistakenly bundled with a group of 103 rejections, thereby making claims 17, 23, 25, and 28 also look as if they were rejected under 35 USC 103. However, from the rejections of each of these claims, it is clear that these are in fact 102 rejections (i.e., each element is pointed out within a single reference and no missing element and motivation to add a missing element exists). Therefore, in this action, these rejections were simply moved to the 102 rejection section. This has no affect on applicant's arguments as applicant argues that Fernando (the 102 reference) does not anticipate the claims.

Conclusion

70. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
December 8, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100